CLAIMS

What is claimed is:

1. A method for allocating address space for a computer platform, comprising:

gathering resource requests for a plurality of peripheral devices hosted by the computer platform;

determining a resource allocation scheme to support the resource requests of the peripheral devices that consumes a minimum amount of address space; and

allocating address space for respective peripheral devices based on the resource allocation scheme that is determined.

2. The method of claim 1, wherein the peripheral devices comprise PCI (Peripheral Component Interconnect) devices.

3. The method of claim 2, wherein the resource allocation scheme is implemented via operations including:

aggregating the resource requests for PCI devices at a given level of a PCI hierarchy for the computer platform into respective resource request objects, each resource request object having a size corresponding to the aggregated resource requests of the PCI devices to which it corresponds;

defining a bin size comprising an address space aperture corresponding to a resource type of the resource requests; and

sorting, via a bin-packing algorithm, the resource request objects into appropriate bins to minimize the number of bins required to support the resource requests for all of the PCI devices hosted by the computer platform.

4. The method of claim 3, wherein the bin-packing algorithm is the K^{th} approximation knapsack algorithm.

5. The method of claim 3, wherein the resource requests are aggregated at the PCI root bridge level.

6. The method of claim 1, wherein the resource requests pertain to peripheral device input/output (I/O) address requests.

7. The method of claim 6, wherein the peripheral device I/O address requests are allocated to a portion of platform address space containing virtual addresses.

8. The method of claim 1, wherein the resource requests pertain to memory onboard peripheral devices that is requested to be mapped into the computer platform address space.

9. The method of claim 1, further comprising determining resource alignment requirements for the resource allocation.

10. The method of claim 1, further comprising performing legacy aliasing, wherein resources are mapped to the address space in a manner that accounts for legacy device addressing considerations.

11. The method of claim 1, further comprising allocating a reserved portion of address space for hot-plug devices.

12. The method of claim 11, wherein the allocation of the reserved portion of address space for hot-plug devices enables dynamic reallocation of resources in response to the removal or addition of a hot-plug device to the computer platform.

13. A method comprising:

determining a PCI (Peripheral Component Interconnect) hierarchy corresponding to PCI devices hosted by a computer platform;

walking the PCI hierarchy to gather resource requests for the PCI devices;

building a map of resource alignment requirements for the PCI devices and bridges;

computing a virtual resource map the maps resource requests to resource allocations in a manner that minimizes the amount of address space in which the resource allocations will fit based on the resource alignment requirements and sizes of the resource requests; and

allocating resources for the PCI devices based on the virtual resource map.

14. The method of claim 13, wherein the algorithm comprises a bin-packing algorithm, and the resources are allocated by performing operations including:

aggregating the resource requests for PCI devices at a given level of the PCI hierarchy into respective resource request objects, each resource request object having a size corresponding to the aggregated resource requests of the PCI devices to which it corresponds;

defining a bin size comprising an address space aperture corresponding to a resource type of the resource requests; and

sorting, via the bin-packing algorithm, the resource request objects into appropriate bins to minimize the number of bins required to support the resource requests for all of the PCI devices hosted by the computer platform.

15. The method of claim 13, wherein the resource requests pertain to PCI input/output (I/O) address requests.

16. The method of claim 13, wherein the resource requests pertain to PCI memory onboard PCI devices that are mapped into the computer platform address space via the resource allocation.

17. The method of claim 13, further comprising allocating a reserved portion of address space for hot-plug devices.

18. The method of claim 13, wherein the resource request objects correspond to resource requests aggregations at a PCI root bridge level.

19. The method of claim 13, wherein the resource request objects correspond to resource request aggregations at a hierarchy level below a PCI root bridge level.

20. The method of claim 13, further comprising performing legacy aliasing, wherein resources are mapped to the address space in a manner that accounts for legacy device addressing considerations.

21. A machine-readable medium via which instructions are provided, which when executed perform operations comprising:

receiving a set of resource requests for peripheral devices in a computer platform;

building a map of resource alignment requirements corresponding to the resource requests;

computing a virtual resource map the maps resource requests to address space resource allocations based on the resource alignment requirements, wherein the address space consumed by the resource requests is minimized; and

allocating address space resources for the peripheral devices based on the virtual resource map.

22. The machine-readable medium of claim 21, wherein the peripheral devices comprise PCI (Peripheral Component Interconnect) devices, and the resource requests correspond to PCI input/output I/O address requests.

23. The machine-readable medium of claim 21, wherein the peripheral devices comprise PCI (Peripheral Component Interconnect) devices, and the resource requests correspond to PCI memory address requests.

24. The machine-readable medium of claim 21, wherein the peripheral devices comprise PCI (Peripheral Component Interconnect) devices, and wherein the virtual resource map is built via operations including:

aggregating the resource requests for PCI devices at a given level of a PCI hierarchy for the computer platform into respective resource request objects, each resource request object having a size corresponding to the aggregated resource requests of the PCI devices to which it corresponds in consideration of the resource alignment requirements for those PCI devices;

sorting, via a bin-packing algorithm, the resource request objects into appropriate bins to minimize the number of bins required to support the resource requests for all of the PCI devices hosted by the computer platform, the bins having a size corresponding to an address space aperture for a resource type of the resource requests.

25. The machine-readable medium of claim 21, wherein execution of the instructions further performs the operation of allocating a reserved portion of address space for hot-plug devices.

26. A system comprising:

a processor;

system memory operatively coupled to the processor;

a plurality of buses, operatively coupled to the processor, including at least one PCI (Peripheral Component Interconnect) bus;

a plurality of PCI devices operatively coupled to said at least one PCI bus; and

a firmware storage device, operatively coupled to the processor, on which firmware instructions are stored, which when executed on the processor perform operations including:

enumerating the PCI devices;

receiving a set of resource requests for respective PCI devices;

building a map of resource alignment requirements corresponding to the resource requests;

computing a virtual resource map that maps resource requests to address space resource allocations based on the resource alignment requirements, wherein the address space consumed by the resource requests is minimized; and

allocating address space resources for the PCI devices based on the virtual resource map.

27. The system of claim 26, wherein the resource requests correspond to PCI input/output (I/O) address requests.

28. The system of claim 26, wherein the resource requests correspond to PCI memory address requests.

29. The system of claim 26, wherein the virtual resource map is built via operations including:

aggregating the resource requests for PCI devices at a given level of a PCI hierarchy for the computer platform into respective resource request objects, each resource request object having a size corresponding to the aggregated resource requests of the PCI devices to which it corresponds in consideration of the resource alignment requirements for those PCI devices;

sorting, via a bin-packing algorithm, the resource request objects into appropriate bins to minimize the number of bins required to support the resource requests for all of the PCI devices in the system, the bins having a size corresponding to an address space aperture for a resource type of the resource requests.

30. The system of claim 29, wherein the bin-packing algorithm is employed for packing two types of resource requests, including PCI input/output (I/O) address requests and PCI memory address requests.